DIME Dynamic Documentation Training Stata Exercise

Luiza Andrade & Mrijan Rimal

June 8, 2017

Introduction

This exercise introduces you to how to export files from Stata that can be read in LaTeX. See exercises 1 and 2 for instructions on how to import files into LaTeX. After this exercise and exercise 1, you will have a document that is automatically updated each time you run your Stata and your LaTeX code.

We have provided you with a do-file that creates files you can import to a LaTeX document. Note that this is not an exercise on creating graphs and tables in Stata, only on exporting Stata outputs into format that LaTeX can read, so this exercise assumes knowledge of some intermediate level Stata commands.

Part 1: Setting up a Folder Structure

Do not rush over this part. You will be forced to write an unecessarily complicated LaTeX code unless you set up a simple but well organized folder structure where you'll save tables and figures created in Stata. We strongly recommend that you start by setting up the following folder structure.

Create one folder called `Output`, and inside this folder, create one folder called `Raw` and one folder called `Final`. You will export tables and graphs from Stata to the `Raw` folder and then import them to your LaTeX document. When you start a new LaTeX, you must save the `.tex` file in the `Final` folder before compiling it, so LaTeX knows where the file paths specified lead.

See the example below. As your project grows bigger, it is common that sub-folders are added in the `Raw` folder. For example, tables and figures often have separate folders. You will find your preferred way to organize this.

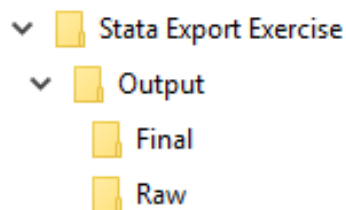Figure 1: Common Folder Structure

Part 2: Setting your path for Stata

We now need to tell Stata to save the tables and figures they export to the `Raw` folder. Using a global with the file path to this folder instead of typing out its location in all commands makes the code simpler and easier to update if you happened to make changes to the folder structure.

You will find a do-file called `Export tables and images.do` in the same folder as this handout. Open it and look for the part that says:

```
global main_folder "<<<ENTER YOUR FOLDER PATH HERE>>>"
```

You need to enter the path to the folder you created in part 1 here. Assuming you are using Windows and following the folder structure in figure 1, this means that the global called `main_folder` should point to your *Stata Export Exercise* folder.[1] If you don't know how to find the file path to your folder, there are two annexes to this document showing how to do it. Annex 1 gives advice for Windows, and Annex 2, for Mac.

Part 3: File Formats and Names

Not all file formats that you can chose when exporting from Stata can be imported into LaTeX. Figures must be saved in `.png` format. All figures produced in Stata can be exported in `.png`. Even if some commands are not able to export in this format, you can use Stata to convert the figure for you. You will find more details on this below, when we export figures.

Tables must be saved in `.tex` format. Tables cannot be converted as easily as figures, so it is usually easier to find an alternative command that produces the same table but can export to `.tex` format. Most commands that export tables are able to export to this format so it should not be difficult to find an alternative.

Files saved in `MS Word`, `MS Excel`, or Stata's `.gph` format cannot imported to LaTeX.

Files exported from Stata should have a descriptive name. Otherwise, you increase the risk for human errors, and the whole point of dynamic documents is to reduce exactly that. For example – rather than `graph1.png`, `graph2.tex`, names like `treatmentEffectGraph.png` would make it easier to understand what files we are using.

Part 4: Install packages and load data

1. Install Stata package *iebaltab*. This package will be used to create and export balance tables.

2. Install Stata package *estout*. This package will be used to export tables created by different Stata commands.

3. Run lines 57 to 81 of do file `Export tables and images.do`. This will load the life expectancy data set built in Stata and adapt it to mimic an experimental data set. As this is not an exercise in how to clean data in Stata, we will not discuss this portion of the do-file.

Part 5, Task 1: Tabulate Categorical Variables

First we want to start by exporting a simple frequencies table for a categorical variable.

1. You should always start a new table export by typing `estimates clear`. This will clear any results already in memory so you only export the desired content.

2. Use the command `tabulate` to generate the frenquency table for variable `region`.

---

[1]Note that the global created in line 34 will only lead to the right folder if you create the folders *Output* and *Raw* manually.

3. Type `estpost` before tabulate to adjust the result of the tabulation to an easily exported format. Type `help estpost` for more details on this command.

4. Use `esttab` to export this table to `.tex` format. Remember to always have an explanatory file name. This is an exercise in LaTeX and not in the `estout` package, so we will not go into more details about `esttab` options, but you can look into it's help file for options. The only difference between using `esttab` to export a table to excel and LaTeX is the file name. If you'd type

   `esttab using "$raw_output/categorical.xls", replace`

   to export a table to excel, to export it to LaTeX simply type

   `esttab using "$raw_output/categorical.tex", replace`

5. Check your `Raw` folder to see the `categorical.tex` file you have exported.

Part 5, Task 2: Balance table

Now we will use *iebaltab* to export a balance table. You can look at the balance table section of do file `Export tables and images.do` to see a basic code using this command. To export this table to LaTeX, you should use option `savetex()` and add the desired file path. Check your output folder to see that the file was created.

Part 5, Task 3: Regression table

1. Clear any results already in memory by typing `estimates clear`

2. Run the regressions whose results you want to export. You can use any regression options allowed by Stata. If Stata can run the regression and display its results in the Stata window, then `esttab` will be able to export them to `.tex` format. In our example do-file, we regressed life expectancy on treatment status and GNP per capita, with and without fixed effects. The difference between the regressions in lines 109 and 113 is that the second one includes region fixed effects. We can use the command `estadd` to add text to the tables. In this example, we use it to indicate the inclusion of fixed effects. See the help file for `estadd` for more details on how this works.

3. Type `eststo:` in the beginning of the line where your regression is. This command stores regressions' result so that `esttab` can export a table with results for more then one regression at a time.

4. Export the stored regression results using `esttab` to LaTeX format by setting the file extension to `.tex`. Like this:

   `esttab using "$raw_output/regression_table.tex", replace`

   `esttab` will export all results stored in memory as estimates. That is why it is important that we start each task with the code `estimates clear`: otherwise we might add results from previous tables to this table.

5. Now check your `Raw` folder to see the `regression_table.tex` file you have exported.

Part 6, Task 1: Manually Create a Graph and then Export it

Stata's save graph feature saves the graph in `.gph` format, which only Stata can read. However, the `graph export` feature saves the graph in a picture format, i.e. png format, which can be read by the photo viewer on your computer, phone and also by LaTeX. It is *absolutely critical* to export the graph to this format, so that LaTeX can import it.

To export a graph as a `.png` file, you need to use the `graph export` command and define PNG as the desired format. In our example do-file, this is done in line 152:

```
graph export "$raw_output/regular_graph.png", width5000 replace
```

Now check your `Raw` folder to see the `regular_graph.png` file you have exported.

*Note: This is different from the **iegraph** command, where save can both export the graph in `.gph` and `.png` format. For Stata's **graph** command, export has be to be used to make it readable in LaTeX.*

Part 6, Task 2: Using `iegraph` to create a figure

Stata's `graph twoway` uses the `save` feature to export pictures in a `.gph` format, and we have to use `graph export` to export it to `.png` format. However, many commands, `iegraph` for example, can export directly to either format using the same `save` option.

When you use `iegraph` to export a plot to LaTeX, always make sure that the picture has the extension `.png` at the end of the filename. **Without that, `iegraph` is just going to export the picture in a format which only Stata can read!**

Using `save("$raw_output/iegraph.png")` ensures that the graphs are directly saved to the specified output folder.

Now check your `Raw` folder to see the `iegraph.png` file you have exported.

Part 7, Using a do-file to edit a .tex file after exporting it

Note: This is an advanced exercise and you don't have to feel compelled to finish it. This is also best done after finishing all the other exercises in the GitHub repository.

During this part of the exercise, you will learn how to use commands in Stata to format your tables. While tables exported from Stata to LaTeX are generally very well formated, sometimes they need to be tweaked a little to look nicer.

1. Run lines 177 to 189 of do-file Add path and name of do-file. This will create a table with sample sizes

for control and treatment groups across regions and in the whole sample. Add this table to the .tex file you created in exercise 1. How does that look?

2. Open file *samplesizes.tex* created by Stata. Can you identify the source of extra spacing?

3. Use command `filefilter` in Stata to filter out the lines or characters in the fragmented file that create the extra spacing. Import the new `.tex` file and check how it looks.

4. Repeat task 3 if necessary.

Annex 1 - Finding Path on a Windows Computer

Option 1 - Copy as path

Path to a file can be found by selecting a file and pressing SHIFT on your keyboard and RIGHT CLICKING your mouse and then clicking COPY AS PATH in the resulting drop down menu as shown in Figure 2.
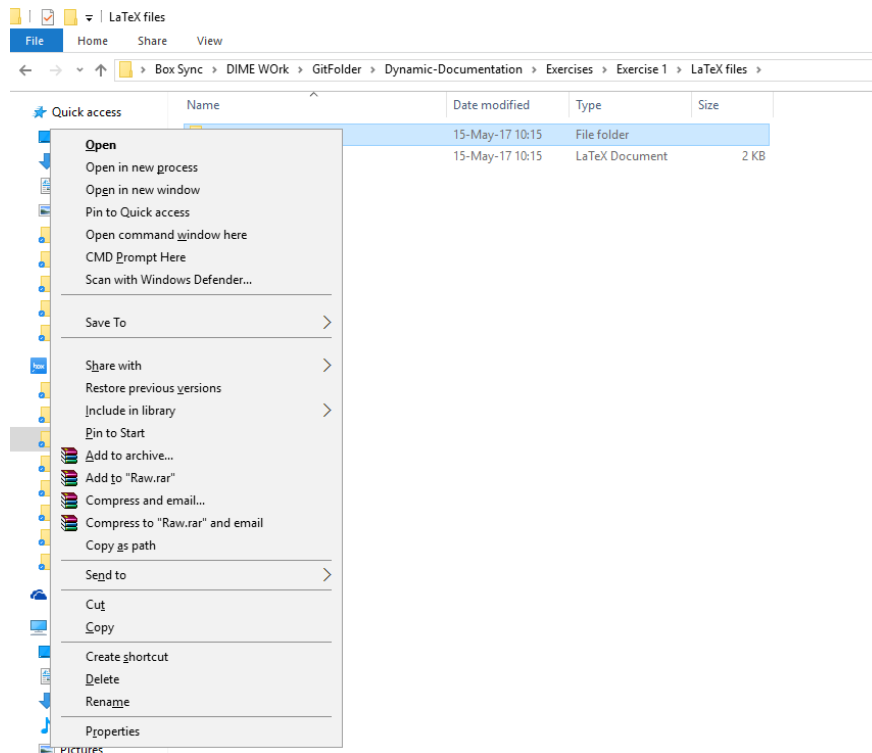

Figure 2: Finding path on a Windows Computer - Solution 1
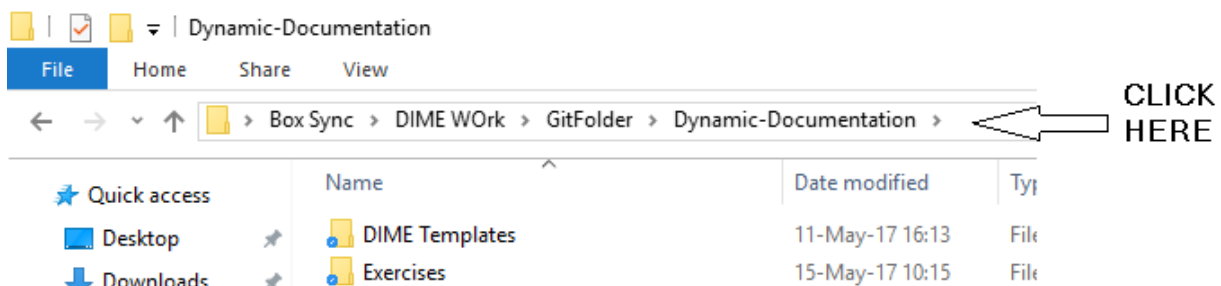
Option 2 - Copy as path

Figure 3: Finding Path on a Windows Computer

Clicking on the bar at the top of the `File Explorer` windows where our files are saved shows us the complete path to the files in a Windows computer.
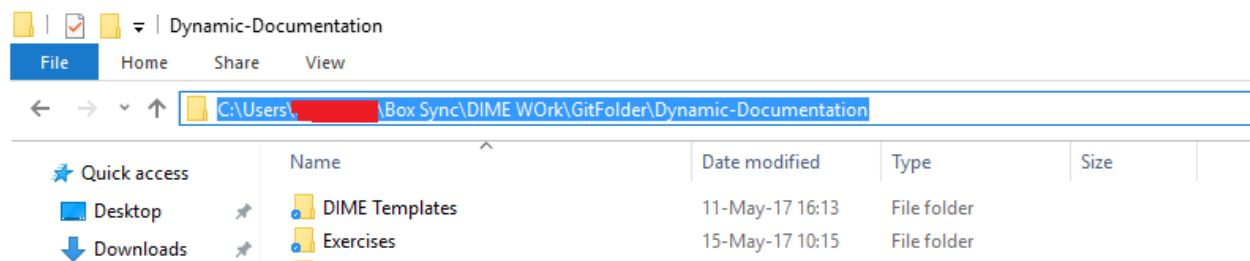
Figure 4: Path shown on a Windows Computer

Once we click on the bar, the complete path to the folder is displayed, as shown in Figure 4. We can then paste this path when setting the path in our Stata do-file and changing the path where it says

```
global main_folder ``<<<ENTER YOUR FOLDER PATH HERE>>>''
```

Annex 2 - Finding Path on a Mac

To copy the path on a Mac, please follow the following steps:

- Right click(or Control-click or two finger click) on the file or folder you want to copy the path of.

- While the right click menu appears, press the `Options` key to reveal ``Copy file as pathname`` button. This is shown in figure 5.

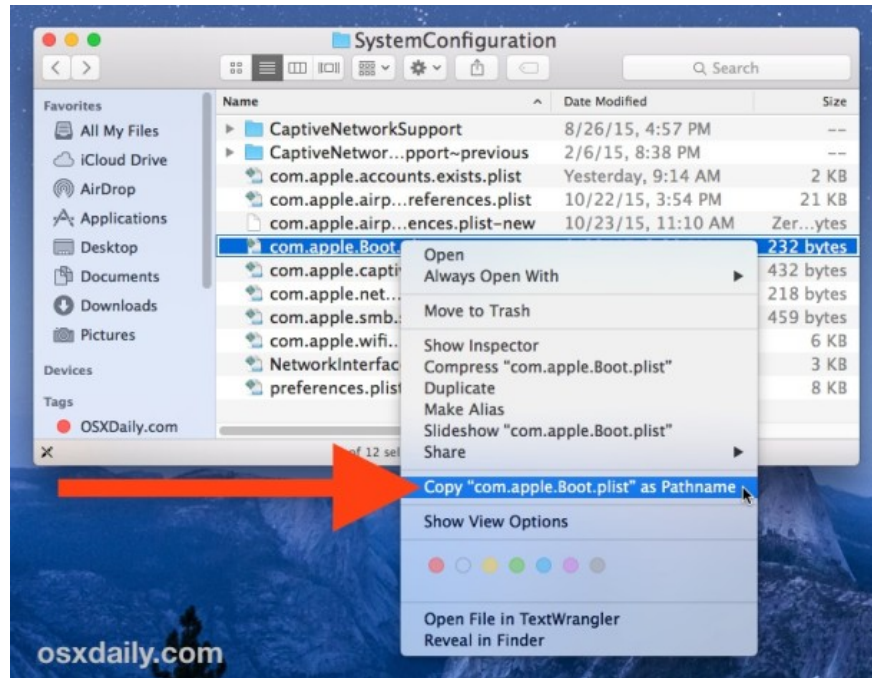- Click on that button and paste anywhere to paste the path name.

Figure 5: Finding a file path on a mac. Photo courtesy osxdaily.com

If you're using an older MAC computer and the option to "Copy file as pathname" does not show up, please follow the following steps:

- Right click(or control-click or two finger click) on your file or folder you want to copy the path of.

- Click on `Get Info` option that appears in the drop down menu.

- In the `Info` window, copy the part that appears after **Where:** and before **Created:**.

- Paste that part anywhere to paste the path name.